

计算机学院专业课程

---

# 计算机组成

## 多周期处理器 形式建模综合方法

高小鹏

北京航空航天大学计算机学院

# 多周期数据通路描述表

- 多周期建模方法与单周期思路完全一致
- 将IR/A/B/ER/ALUOut/DR插入到表格中
  - ◆ 注意：这些寄存的输入是固定的
- 因为插入了寄存器，改变了部件间连接关系

| 部件     | 输入  | 指令     |
|--------|-----|--------|
| NPC    | PC  |        |
|        | Imm |        |
| PC     | NPC |        |
| IM     | Ad  |        |
| IR     |     | IM     |
| RF     | WD  |        |
|        | A3  |        |
| A      |     | RF.RD1 |
| B      |     | RF.RD2 |
| EXT    | Imm |        |
| ER     |     | EXT    |
| ALU    | A   |        |
|        | B   |        |
| ALUOut |     | ALUOut |
| DM     | Ad  |        |
|        | Din |        |
| DR     |     | DM.DO  |

# lw: 从RTL到数据通路

- 部分部件的输入源改为寄存器（注意红色信号）

| 部件     | 输入  | lw        |
|--------|-----|-----------|
| NPC    | PC  | PC.PC     |
|        | Imm |           |
| PC     | NPC | NPC.NPC   |
| IM     | Ad  | PC.PC     |
| IR     |     | IM        |
| RF     | WD  | DR        |
|        | A3  | IR[20:16] |
| A      |     | RF.RD1    |
| B      |     | RF.RD2    |
| EXT    | Imm | IR[15:0]  |
| ER     |     | EXT       |
| ALU    | A   | A         |
|        | B   | EXT       |
| ALUOut |     | ALU       |
| DM     | Ad  | ALUOut    |
|        | Din |           |
| DR     |     | DM.DO     |

| lw        |
|-----------|
|           |
|           |
|           |
|           |
|           |
|           |
| DM.DO     |
| IM[20:16] |
|           |
|           |
|           |
| IM[15:0]  |
| EXT.Ext   |
| RF.RD1    |
| EXT.Ext   |
|           |
| ALU.C     |
|           |
|           |

单周期

| 周期 | RTL   | 控制信号                                    |
|----|---|---|
| 1  | $IR \leftarrow IR[PC];$<br>$PC \leftarrow NPC(PC)$  | $IRWr: 1;$<br>$NPCOp: +4;$<br>$PCWr: 1$ |
| 2  | $A \leftarrow RF[rs]$<br>$ER \leftarrow EXT(Imm16)$ | $EXTOp: SE$                             |
| 3  | $ALUOut \leftarrow ALU(A, EXT)$                     | $ALUOp: ADD$                            |
| 4  | $DR \leftarrow DM[ALUOut]$                          |   |
| 5  | $RF[rt] \leftarrow DR$                              | $RFWr: 1$                               |



# jal: 从RTL到数据通路

| 部件     | 输入  | jal      |
|--------|-----|----------|
| NPC    | PC  | PC.PC    |
|        | Imm | IR[25:0] |
| PC     | NPC | NPC.NPC  |
| IM     | Ad  | PC.PC    |
| IR     |     | IM       |
| RF     | WD  | PC.PC    |
|        | A3  | 0x1F     |
| A      |     |          |
| B      |     |          |
| EXT    | Imm |          |
| ER     |     |          |
| ALU    | A   |          |
|        | B   |          |
| ALUOut |     |          |
| DM     | Ad  |          |
|        | Din |          |
| DR     |     |          |

TIP:

在单周期中，RF的回写数据来源于NPC.PC4。

在多周期中，由于PC已经在cycle 1完成了PC+4计算，因此我们不再需要NPC.PC4了。

| 周期 | RTL  | 控制信号   |
|----|--|--|
| 1  | $IR \leftarrow IM[PC];$<br>$PC \leftarrow NPC(PC)$       | <b>IRWr: 1</b><br><b>NPCOp: +4</b><br><b>PCWr: 1</b>   |
| 2  | $RF[31] \leftarrow PC$<br>$PC \leftarrow NPC(PC, imm26)$ | <b>RFWr: 1</b><br><b>NPCOp: JNPC</b><br><b>PCWr: 1</b> |



# 用同样方法建模 其他指令的多周期数据通路



# 综合形成完整的数据通路

| 部件     | 输入  | addu      | addiu     | beq      | lw        | sw       | jal      | 综合                             |
|--------|-----|-----------|-----------|----------|-----------|----------|----------|--------------------------------|
| NPC    | PC  | PC.PC     | PC.PC     | PC.PC    | PC.PC     | PC.PC    | PC.PC    | PC.PC                          |
|        | Imm |           |           | IR[15:0] |           |          | IR[25:0] | IR[25:0]                       |
| PC     | NPC | NPC.NPC   | NPC.NPC   | NPC.NPC  | NPC.NPC   | NPC.NPC  | NPC.NPC  | NPC.NPC                        |
| IM     | Ad  | PC.PC     | PC.PC     | PC.PC    | PC.PC     | PC.PC    | PC.PC    | PC.PC                          |
| IR     |     | IM        | IM        | IM       | IM        | IM       | IM       | IM                             |
| RF     | WD  | ALUOut    | ALUOut    |          | DR        |          | PC       | ALUOut<br>DM.DO<br>NPC.PC4     |
|        | A3  | IR[15:11] | IR[20:16] |          | IR[20:16] |          | 0x1F     | IR[15:11]<br>IR[20:16]<br>0x1F |
| A      |     | RF.RD1    | RF.RD1    | RF.RD1   | RF.RD1    | RF.RD1   |          | RF.RD1                         |
| B      |     | RF.RD2    |           | RF.RD2   |           |          |          | RF.RD2                         |
| EXT    | Imm |           | IR[15:0]  |          | IR[15:0]  | IR[15:0] |          | IR[15:0]                       |
| ER     |     |           | EXT.Ext   |          | EXT.Ext   | EXT.Ext  |          | EXT.Ext                        |
| ALU    | A   | A         | A         | A        | A         | A        |          | A                              |
|        | B   | B         | ER        | B        | ER        | ER       |          | B<br>ER                        |
| ALUOut |     | ALU       | ALU       |          | ALU       | ALU      |          | ALU                            |
| DM     | Ad  |           |           |          | ALUOut    | ALUOut   |          | ALUOut                         |
|        | Din |           |           |          |           | B        |          | B                              |
| DR     |     |           |           |          | DM.DO     |          |          | DM.DO                          |

# 构造MUX

| 部件     | 输入  | addu      | addiu     | beq      | lw        | sw       | jal      | 综合                             |
|--------|-----|-----------|-----------|----------|-----------|----------|----------|--------------------------------|
| NPC    | PC  | PC.PC     | PC.PC     | PC.PC    | PC.PC     | PC.PC    | PC.PC    | PC.PC                          |
|        | Imm |           |           | IR[15:0] |           |          | IR[25:0] | IR[25:0]                       |
| PC     | NPC | NPC.NPC   | NPC.NPC   | NPC.NPC  | NPC.NPC   | NPC.NPC  | NPC.NPC  | NPC.NPC                        |
| IM     | Ad  | PC.PC     | PC.PC     | PC.PC    | PC.PC     | PC.PC    | PC.PC    | PC.PC                          |
| IR     |     | IM        | IM        | IM       | IM        | IM       | IM       | IM                             |
| RF     | WD  | ALUOut    | ALUOut    |          | DR        |          | PC       | ALUOut<br>DR<br>PC             |
|        | A3  | IR[15:11] | IR[20:16] |          | IR[20:16] |          | 0x1F     | IR[15:11]<br>IR[20:16]<br>0x1F |
| A      |     | RF.RD1    | RF.RD1    | RF.RD1   | RF.RD1    | RF.RD1   |          | RF.RD1                         |
| B      |     | RF.RD2    |           | RF.RD2   |           |          |          | RF.RD2                         |
| EXT    | Imm |           | IR[15:0]  |          | IR[15:0]  | IR[15:0] |          | IR[15:0]                       |
| ER     |     |           | EXT.Ext   |          | EXT.Ext   | EXT.Ext  |          | EXT.Ext                        |
| ALU    | A   | A         | A         | A        | A         | A        |          | A                              |
|        | B   | B         | ER        | B        | ER        | ER       |          | B<br>ER                        |
| ALUOut |     | ALU       | ALU       |          | ALU       | ALU      |          | ALU                            |
| DM     | Ad  |           |           |          | ALUOut    | ALUOut   |          | ALUOut                         |
|        | Din |           |           |          |           | B        |          | B                              |
| DR     |     |           |           |          | DM.DO     |          |          | DM.DO                          |

MRFWDD

MRFA3

MALUB

| 部件  | 输入  | addu    | addiu   | beq      | lw      | sw      | jal      | 综合                 |
|-----|-----|---------|---------|----------|---------|---------|----------|--------------------|
| NPC | PC  | PC.PC   | PC.PC   | PC.PC    | PC.PC   | PC.PC   | PC.PC    | PC.PC              |
|     | Imm |         |         | IR[15:0] |         |         | IR[25:0] | IR[25:0]           |
| PC  | NPC | NPC.NPC | NPC.NPC | NPC.NPC  | NPC.NPC | NPC.NPC | NPC.NPC  | NPC.NPC            |
| IM  | Ad  | PC.PC   | PC.PC   | PC.PC    | PC.PC   | PC.PC   | PC.PC    | PC.PC              |
| IR  |     | IM      | IM      | IM       | IM      | IM      | IM       | IM                 |
| RF  | WD  | ALUOut  | ALUOut  |          | DR      |         | PC       | ALUOut<br>DR<br>PC |
|     |     |         |         |          |         |         |          | IR[15:11]          |

MRFWD

## 建立指令与MUX控制信号的取值对应关系

示例：由于ALUOut位于MRFWD的0通道，因此addu和addiu对应的控制信号选择就应为0

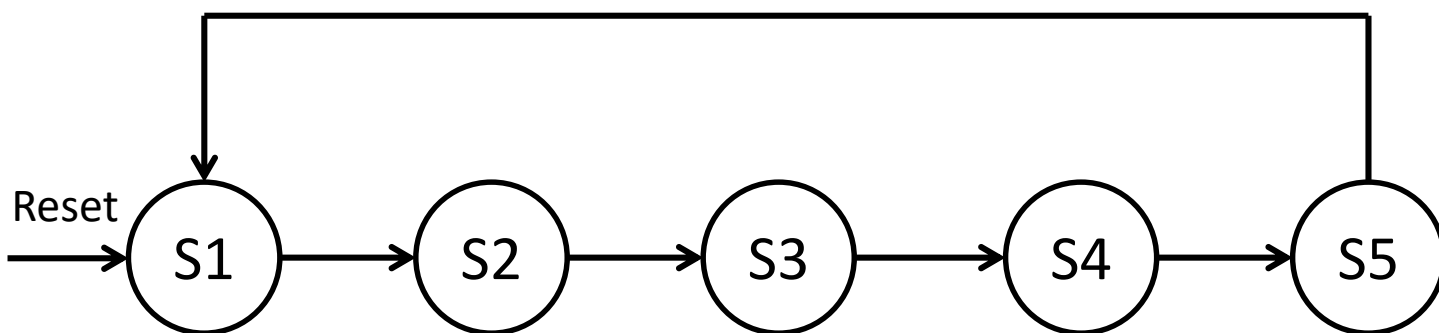
TIP：建议在Verilog中使用宏，而不是直接用00,01,10  
例如：`ALU~2'b00, `MEM~2'b01, `PC~2'b10

|    |  |      |      |    |       |      |     |        |
|----|--|------|------|----|-------|------|-----|--------|
| DR |  |      |      |    | DM DO |      |     | DM DO  |
|    |  | `ALU | `ALU |    | `MEM  |      | `PC | MRFWD  |
|    |  | RD   | RD   |    | RI    |      | R31 | IMRFA3 |
|    |  | `B   | `B   | `B | `EXT  | `EXT |     | MALUB  |



# lw: 状态机及控制信号

- 状态数: RTL表的周期数决定指令的执行状态数
- 控制信号: 查RTL表&数据通路表确定取值
  - 注意颜色之间的关联关系



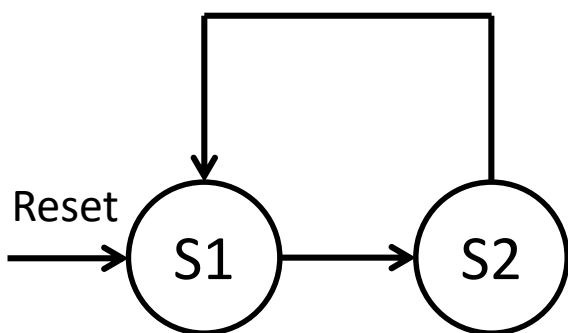
|      |       |     |      |      |      |       |
|------|-------|-----|------|------|------|-------|
| addu | addiu | beq | lw   | sw   | jal  | 综合    |
| `ALU | `ALU  |     | `MEM |      | `PC  | MRFW  |
| `RD  | `RD   |     | `RT  |      | `R31 | MRFA3 |
| `B   | `B    | `B  | `EXT | `EXT |      | MALUB |

| 周期 | 控制信号                              |
|----|-----------------------------------|
| 1  | IRWr: 1;<br>NPCOp: +4;<br>PCWr: 1 |
| 2  | EXTOp: SE                         |
| 3  | ALUOp: ADD                        |
| 4  |                                   |
| 5  | RFWr: 1                           |

|    | NPCOp    | PCWr | IRWr | RFWr | DMWr | EXTOp   | ALUOp    | MRFW | MRFA3 | MALUB |
|----|----------|------|------|------|------|---------|----------|------|-------|-------|
| lw | S1: `PC4 | S1   | S1   | S5   |      | S2: `SE | S3: `ADD | `MEM | `RT   | `EXT  |

# jal: 状态机及控制信号

- 由于jal只有2个cycle，因此从S2返回



|      |       |     |      |      |      |       |
|------|-------|-----|------|------|------|-------|
| addu | addiu | beq | lw   | sw   | jal  | 综合    |
| `ALU | `ALU  |     | `MEM |      | `PC  | MRFWD |
| `RD  | `RD   |     | `RT  |      | `R31 | MRFA3 |
| `B   | `B    | `B  | `EXT | `EXT |      | MALUB |

| 周期 | 控制信号                              |
|----|-----------------------------------|
| 1  | IRWr: 1<br>NPCOp: +4<br>PCWr: 1   |
| 2  | RFWr: 1<br>NPCOp: JNPC<br>PCWr: 1 |

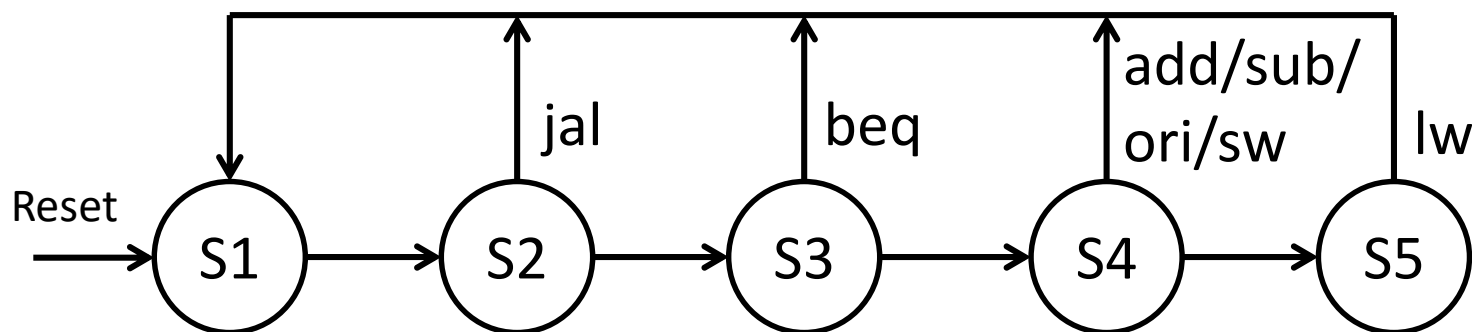
|     | NPCOp    | PCWr | IRWr | RFWr | DMWr | EXTOp | ALUOp | MRFWD | MRFA3 | MALUB |
|-----|----------|------|------|------|------|-------|-------|-------|-------|-------|
| jal | S1: `PC4 | S1   | S1   | S2   |      |       |       | `MEM  | `RT   |       |
|     | S2: JNPC | S2   |      |      |      |       |       |       |       |       |

# 用同样方法建模 其他指令的状态机及控制信号



# 合成：状态机及控制信号

- 状态机合成：只需要增加状态返回初态（S1）的**判断条件**
  - ◆ 判断条件就是指令变量
  - ◆ 指令变量的产生方法与单周期控制器完全相同



# 合成：状态机及控制信号

## 控制信号合成

注：对应MUX的控制信号

|      | NPCOp               | PCWr          | IRWr | RFWr | DMWr | EXTOp  | ALUOp   | MRFWD   | MRFA3 | MALUB |
|------|---------------------|---------------|------|------|------|--------|---------|---------|-------|-------|
| lw   | S1:`PC4             | S1            | S1   | S5   |      | S2:`SE | S3:`ADD | `ALUOUT | `RT   | `EXT  |
| sw   | S1:`PC4             | S1            | S1   | 0    | S4   | S2:`SE | S3:`ADD | `ALUOUT | `RT   | `EXT  |
| addu | S1:`PC4             | S1            | S1   | S4   |      |        | S3:`ADD | `ALUOUT | `RD   | `B    |
| sub  | S1:`PC4             | S1            | S1   | S4   |      |        | S3:`SUB | `ALUOUT | `RD   | `B    |
| ori  | S1:`PC4             | S1            | S1   | S4   |      | S2:`ZE | S3:`OR  | `ALUOUT | `RT   | `EXT  |
| lui  | S1:`PC4             | S1            | S1   | S3   |      | S2:`HE |         | `ER     | `RT   |       |
| beq  | S1:`PC4<br>S3:`BNPC | S1<br>S3:Zero | S1   |      |      |        |         |         |       |       |
| jal  | S1:`PC4<br>S2:`JNPC | S1<br>S2      | S1   | S2   |      |        |         | `PC     | `R31  |       |

## 表达式构造与上讲方法相同

$$\begin{aligned} \text{PCWr} = & (\text{lw} + \text{sw} + \text{addu} + \text{subu} + \text{ori} + \text{lui} + \text{beq} + \text{jal}) \cdot \text{S1} + \\ & \text{beq} \cdot \text{Zero} \cdot \text{S3} + \\ & \text{jal} \cdot \text{S2} \end{aligned}$$

由于S1是所有指令的公共状态，可以优化为

$$\text{PCWr} = \text{S1} + \text{beq} \cdot \text{Zero} \cdot \text{S3} + \text{jal} \cdot \text{S2}$$

# 合成：状态机及控制信号

$ALUOp = (lw+sw+addu) ? S3 \cdot \text{`ADD} :$   
 $(sub) \quad \quad \quad ? S3 \cdot \text{`SUB} :$   
 $\quad \quad \quad \quad \quad S3 \cdot \text{`OR}$

□ 非使能类信号：如果只在同一周期有效，则可以优化掉时间信息

◆ 全局时间范围内有效也是没有错误的：硬件的“尽早执行”原则

$ALUOp = (lw+sw+addu) ? \text{`ADD} :$   
 $(sub) \quad \quad \quad ? \text{`SUB} :$   
 $\quad \quad \quad \quad \quad \text{`OR}$

Q：如果一个非使能类控制信号在不同周期有效，则不能优化掉时间信息。为什么？

|      | NPCOp                 | PCWr           | IRWr | RFWr | DMWr | EXTOp   | ALUOp    |         |      |      |
|------|-----------------------|----------------|------|------|------|---------|----------|---------|------|------|
| lw   | S1: `PC4              | S1             | S1   | S5   |      | S2: `SE | S3: `ADD |         |      |      |
| sw   | S1: `PC4              | S1             | S1   | 0    | S4   | S2: `SE | S3: `ADD |         |      |      |
| addu | S1: `PC4              | S1             | S1   | S4   |      |         | S3: `ADD |         |      |      |
| sub  | S1: `PC4              | S1             | S1   | S4   |      |         | S3: `SUB | `ALUOUT | `RD  | `B   |
| ori  | S1: `PC4              | S1             | S1   | S4   |      | S2: `ZE | S3: `OR  | `ALUOUT | `RT  | `EXT |
| lui  | S1: `PC4              | S1             | S1   | S3   |      | S2: `HE |          | `ER     | `RT  |      |
| beq  | S1: `PC4<br>S3: `BNPC | S1<br>S3: Zero | S1   |      |      |         |          |         |      |      |
| jal  | S1: `PC4<br>S2: `JNPC | S1<br>S2       | S1   | S2   |      |         |          | `PC     | `R31 |      |

# 执行前转移（PPT）vs 执行后转移（教科书）

- ❑ 复杂度：执行前方案需要精心规划，对开发者要求高
  - ◆ 特别是新增某些状态路径不能轻松区分指令，易导致大幅度修改方案
- ❑ 状态数：数量冗余，却无法优化
  - ◆ S2/S6/S8均为第3个状态，即均可控制ALU，但因分属不同路径而不能合并
- ❑ 实施难度：执行后仅需要建立多周期RTL描述即可
  - ◆ 执行后：从RTL推导状态机和控制器，完全是机械的
  - ◆ 执行前：虽没有显式使用RTL，但事实上隐含了对多周期的分析和划分
- ❑ 性能/资源：前者远高于后者
  - ◆ 指令集规模达到数十条

